

### **Data Analysis Report**

The four group members divided the workload for creating the ERD model by assigning each member to different areas of the given business description.

Malaine Atienza assisted in creating the entities for the rooms, accounting for the variations in a room's purpose and the type of beds a room has that determines its role flexibility. She also helped refine the entities describing the relationship between customers, their charges, and invoices. Given that a room's purpose may vary unless it contains a permanent bed, a general room table was created which may contain data on both sleeping and meeting rooms. A separate table was also created based on bed information that includes the bed size, quantity, and type (permanent, folding, etc.) that a certain room has to help determine its flexibility in being turned into either a meeting or sleeping room. With this, a table was created to store the possible functionality types of a room. To connect this to the room table, a bridging table was created to store data on a room's current functionality. This includes the room's capacity as we assume this may also vary with a room's purpose. Lastly, a table containing the room-adjacency relationships and the rooms involved in each relationship was created. To connect this to the room table, a bridging table was created as one relationship contains many rooms and one room may be part of up to two relationships.

Maya Felix covered the entities for the building information. Including the wing, floor, room, etc. and information about guests, the responsible party, and some of the billing/invoice entities. We are assuming that all reservations are made the same way. We decided not to have a field in the room table to describe the room type and if it is convertible or not because there are multiple types of room function. We use a lookup table to keep track of all the types instead. Rooms can be reassigned during a guest's stay but the reservation is connected to a single roomId - either we don't include this or we just dynamically change room in that one spot instead of adding a new reservation room table etc. We included a room status/occupancy time lookup table. It was not necessary but felt cleaner/better. The room table has information that you can get through foreign keys to other tables but was included anyway because it seemed useful to have there. (smoking status, etc). We didn't include ways to keep track of capacities differently for rooms that are divided. It is unclear how to classify a suite when there is one hallway access but multiple internal rooms – are they booked together or separately? Also, we did not include

ways to adjust billing/reservation when rooms are combined. Rooms can be reassigned during a guest's stay but reservation ties to a single roomId. We don't include a way to keep track of multiple rooms in reservation but you can dynamically change rooms in that one spot instead. Multiple guests in a sleeping room can split billing in any way, but there's a separate charge system for facilities like phones, we did not include this in the billing system.

Emily Elmouaquite covered the entities for events and reservations. Tables were created for events, the room(s) that the events take place in, the guests that attend events, the host (of both reservations and events), and the reservation. Bridging tables were created for the rooms that events take place in, and the guests that attend events in order to connect customers and rooms with events. Assuming that events are booked through their own reservation, eventId was included in the reservation table. Reservations being filled with the lowest numbered wing(s) and the lowest available room number(s) that meets the guests' stated requirements was not addressed in the reservation table because enforcing this rule in the database could limit flexibility in cases where staff or guests have specific preferences that override the default lowest-numbered assignment, and seems more procedural than something that could be implemented in a database. Additionally, reservationStatus refers to whether the customer has confirmed/ checked in or out.

Bianca Nolte covered the entities for registering the billing party, the usage of the facilities by members (guests or employees), the guests' contact and the guest call. The billing party was created under the assumption that only one individual can represent an entity, either an organization or a group of guests, such as a family. As a singular guest can possess multiple invoices, a bridging table was created to connect guest invoices and their respective billing parties. For this bridging table another assumption was formed, that each guest/organization can only have one billing party responsible for them. The usage of the facilities table describes the members' positions throughout the different hotel locations, such as their time of entry and exit. The assumption for this table is that members can only be at one location at a time. We decided to generate separate tables for hotel employees and guests, as they are different data values and they could not belong to the same field. Additionally, the guest contact entity reflects the availability for each guest and each room they can be contacted on. An assumption made here is that each guest can have multiple rooms as points from where they can be contacted by hotel staff. The guest discretion entity shows the confidentiality of each guest, assuming that their

discretion type does not vary throughout their stay unless they request so. The guest call table logs each call received by the hotel, their duration, status, their intended guest and if it should be relayed to them, where we assume that each guest could receive multiple calls during their stay. We did not yet include an entity where the guest is able to make calls.